<u>Amendments to the Claims:</u>

Re-write the claims as set forth below. This listing of claims will replace all prior versions and listings, of claims in the application:

<u>Listing of Claims:</u>

1. (Currently amended) A method for facilitating prevention of interception of incoming data that is provided for an executing software application, comprising:

preventing interception of the incoming data, by providing non key based insertion data for insertion as part of the incoming data;

storing the provided insertion data; [[and]]

filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data; and

controlling timing of insertion data generation and output based on data queue parameters.

2. (Previously presented) The method of claim 1 including processing the actual data resultant from filtering for use by the executing software application.

3. (Previously presented) The method of claim 1 including:

receiving the provided insertion data and actual data from a data input source; and

queuing the insertion data with the actual data for output as the incoming data.

4. (Previously presented) The method of claim 1 including analyzing foreground indication data and enabling generation of the insertion data in response to the foreground indication data.

2

5. – 9. (Canceled)

10.　(Previously presented)　A method for facilitating prevention of interception of incoming data that is provided for an executing software application, comprising:

analyzing foreground indication data and enabling generation of insertion data in response to the foreground indication data;

storing a list of data representing data to be randomized;

selecting data from the list of data as random insertion data that prevents interception of incoming data;

providing selected insertion data for insertion as part of the incoming data;

formatting the insertion data in a same format as actual data;

storing the provided insertion data;

mixing the insertion data with the incoming data; and

filtering received incoming data containing actual data and the random insertion data by comparing stored generated insertion data with incoming data to determine which data is actual data.

11.　(Previously presented)　The method of claim 10 including processing the actual data resultant from filtering for use by the executing software application.

12.　(Previously presented)　The method of claim 10 including:

receiving the provided random insertion data and actual data from a data input source; and

3

queuing the random insertion data with the actual data for output as the incoming data.

13.   (Previously presented) The method of claim 10 including controlling timing of random insertion data generation and output based on data queue parameters.

14.   (Previously presented) The method of claim 10 wherein storing includes storing the generated insertion data as entries in a first in first out (FIFO) buffer and wherein filtering further includes comparing the incoming data to at least one entry of stored insertion data and deleting the at least one entry of stored random data, and deleting the incoming data when the stored data is the same as the incoming data, and outputting the incoming data to a data processor when the stored data from the entry is different from the incoming data.

15.   (Original) The method of claim 14 wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

16.   (Currently amended) An apparatus for facilitating prevention of interception of incoming data comprising:

means for providing non key based insertion data, that prevents interception of the incoming data, for insertion as part of the incoming data;

means for storing the provided insertion data; [[and]]

4

means for filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data for use by a software application;

wherein the means for filtering receives the provided insertion data and actual data, and wherein the apparatus includes means for queuing the insertion data and the actual data for output as the incoming data; and

wherein the means for providing the insertion data includes timing means for controlling timing of random insertion data generation and output based on data queue parameters.

17.   (Canceled)

18.   (Canceled)

19.   (Currently amended) The apparatus of claim [[17]]16 wherein the means for providing insertion data generates the insertion data in response to foreground indication data.

20. – 22. (Canceled)

23.   (Currently amended) The apparatus of claim [[22]]49 wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

5

24. (Previously presented) An apparatus for facilitating prevention of interception of incoming data comprising:

means for storing a list of data representing data to be randomized;

means, operatively coupled to the means for storing the list of data, for randomly selecting data from the list of data

means for providing random insertion data, using the randomly selected data from the list of data, for insertion as part of the incoming data in response to foreground indication data;

means for formatting the random insertion data in a same format as actual data;

means for mixing the insertion data with the actual data;

means for storing the random insertion data; and

means for filtering received incoming data containing actual data and the random insertion data by comparing stored generated random insertion data with incoming data to determine which data is actual data for use by a software application.

25. (Original) The apparatus of claim 24 including means, operatively coupled to the means for filtering, for processing the actual data resultant from filtering for use by the software application.

26. (Original) The apparatus of claim 24 wherein the means for filtering receives the generated random insertion data and actual data, and wherein the apparatus includes means for queuing the random insertion data and the actual data for output as the incoming data.

6

27. (Original) The apparatus of claim 24 wherein the means for generating the random insertion data includes timing means for controlling timing of random insertion data generation and output based on data queue parameters.

28. (Previously presented) The apparatus of claim 24 wherein the means for storing includes a first in first out (FIFO) buffer that stores the generated random insertion data as entries therein and wherein the means for filtering incoming data further includes means for comparing the incoming data to at least one entry of stored random insertion data and deleting the at least one entry of stored random data, and deleting the incoming data when the stored random data is the same as the incoming data, and outputting the incoming data to a data processor when the stored random data from the entry is different from the incoming data.

29. (Previously presented) The apparatus of claim 26 wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

30. (Currently amended) A storage medium comprising memory containing executable instructions that when read by one or more processing units, causes one or more processing units to:

generate non key based insertion data, that prevents interception of incoming data, for insertion as part of the incoming data, for a software application;

store the generated insertion data; [[and]]

filter received incoming data containing actual data and the insertion data by comparing stored generated insertion data with incoming data to determine which data is actual data;

analyze foreground indication data and enable generation of the random insertion data in response to the foreground indication data; and

control timing of random insertion data generation and output based on data queue parameters.

31. (Original) The storage medium of claim 30 containing executable instructions that when read by one or more processing units, causes one or more processing units to process the actual data resultant from filtering for use by the software application.

32. (Original) The storage medium of claim 30 containing executable instructions that when read by one or more processing units, causes one or more processing units to:

receive the generated insertion data and actual data from a data input source; and

queue the insertion data and the actual data for output as the incoming data.

33. – 36. (Canceled)

37. (Currently amended) The storage medium of claim [[36]]51 wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized key stroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

38.   (Previously presented) The method of claim 1 wherein the insertion data is fake data.

39.   (Previously presented) The apparatus of claim 16 wherein the insertion data is fake data.

40.   (Previously presented) The storage medium of claim 30 wherein the insertion data is fake data.

41.   (Canceled)

42.   (Previously presented) A method for facilitating prevention of interception of incoming data that is provided for a software application, comprising:

preventing interception of the incoming data, by providing insertion data for insertion as part of the incoming data;

storing the provided insertion data;

filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data; and

wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

43.   (Canceled)

44. (new) A method for facilitating prevention of interception of incoming data that is provided for an executing software application, comprising:

preventing interception of the incoming data, by providing non key based insertion data for insertion as part of the incoming data;

storing the provided insertion data;

filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data; and

wherein storing includes storing the provided insertion data as entries in a first in first out (FIFO) buffer and wherein filtering further includes comparing the incoming data to at least one entry of stored insertion data and deleting the at least one entry of stored data and deleting the incoming data when the stored random data is the same as the incoming data, and outputting the incoming data to a data processor when the stored random data from the entry is different from the incoming data.

45. (new) A method for facilitating prevention of interception of incoming data that is provided for an executing software application, comprising:

preventing interception of the incoming data, by providing non key based insertion data for insertion as part of the incoming data;

storing the provided insertion data;

filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data;

storing a list of data representing data to be used as randomization data;

CHICAGO/#16233911

randomly selecting the randomized data from the list of data; and

formatting the randomized data as insertion data in a same format as actual data.

46.　(new)　A method for facilitating prevention of interception of incoming data that is provided for an executing software application, comprising:

preventing interception of the incoming data, by providing non key based insertion data for insertion as part of the incoming data;

storing the provided insertion data;

filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data; and

wherein the actual data is keystroke data from a keystroke source, wherein the insertion data is randomized keystroke data and wherein the keystroke data and the randomized keystroke data are queued in a message queue and passed in the form of a message to the software application as incoming data.

47.　(new)　A method for facilitating prevention of interception of incoming data that is provided for an executing software application, comprising:

preventing interception of the incoming data, by providing non key based insertion data for insertion as part of the incoming data;

storing the provided insertion data;

filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data; and

11

wherein preventing includes providing the insertion data, under control of the software application that is to receive the incoming data.

48.	(new)	An apparatus for facilitating prevention of interception of incoming data comprising:

means for providing non key based insertion data, that prevents interception of the incoming data, for insertion as part of the incoming data;

means for storing the provided insertion data;

means for filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data for use by a software application; and

wherein the means for storing includes a first in first out (FIFO) buffer that stores the generated random insertion data as entries therein and wherein the means for filtering incoming data further includes means for comparing the incoming data to at least one entry of stored random insertion data and deleting the at least one entry of stored random data, and deleting the incoming data when the stored random data is the same as the incoming data, and outputting the incoming data to a data processor when the stored random data from the entry is different from the incoming data.

49.	(new)	An apparatus for facilitating prevention of interception of incoming data comprising:

means for providing non key based insertion data, that prevents interception of the incoming data, for insertion as part of the incoming data;

means for storing the provided insertion data;

12

means for filtering received incoming data containing actual data and the insertion data by comparing stored provided insertion data with incoming data to determine which data is actual data for use by a software application;

means, operatively coupled to the means for filtering, for processing the actual data resultant from filtering for use by the software application;

means for storing a list of data representing data to be randomized;

means, operatively coupled to the means for storing the list of data, for selecting data from the list of data as random insertion data; and

means for formatting the random insertion data in a same format as actual data.

50.   (new)   A storage medium comprising memory containing executable instructions that when read by one or more processing units, causes one or more processing units to:

generate non key based insertion data, that prevents interception of incoming data, for insertion as part of the incoming data, for a software application;

store the generated insertion data;

filter received incoming data containing actual data and the insertion data by comparing stored generated insertion data with incoming data to determine which data is actual data; and

store the generated insertion data as entries in a first in first out (FIFO) buffer and compare the incoming data to at least one entry of stored insertion data and delete the at least one entry of stored data, and delete the incoming data when the stored data is the same as the incoming data, and output the incoming data to a data processor when the stored data from the entry is different from the incoming data.

51.   (new)   A storage medium comprising memory containing executable instructions that when read by one or more processing units, causes one or more processing units to:

generate non key based insertion data, that prevents interception of incoming data, for insertion as part of the incoming data, for a software application;

store the generated insertion data;

filter received incoming data containing actual data and the insertion data by comparing stored generated insertion data with incoming data to determine which data is actual data;

store a list of data representing data to be randomized;

randomly select data from the list of data as random insertion data; and

format the random insertion data in a same format as actual data.